CIN

>> operator does not perform bounds checking

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-20

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 3470 bytes

Attack Category	• Malicious Input
Vulnerability Category	• Buffer Overflow
Software Context	• String Management
Location	
Description	The >> operator provided in "istream" does not perform bounds checking on character buffers. If an input is provided that exceeds the length of the character buffer, a buffer overflow will occur.
APIs	**FunctionName**: operator>> ; **Comments**:
Method of Attack	If an attacker controls the input string being processed by a program using the >> operator, he can cause a buffer overflow if a proper call to ios_base::width is not made first. The string provided by the attacker would overwrite other data in the program's memory space, enabling the attacker to crash the application or inject malicious code into the application.
Exception Criteria	When ios_base::width is set to at most the size of the character buffer before the >> operator is used to obtain a string.

Solutions

Solution Applicability	**Solution Description**	**Solution Efficacy**
Whenever operator >> is used to read an input stream into a buffer.	ios_base::width should be set to the size of the character buffer prior to use of the >> operator to read a string. The ios_base::width is reset to zero after each use of	Effective.

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

	the >> operator. Note that ios_base::width is the maximum length of the input string including the null terminator. So at most ios_base::width - 1 printable characters are read into the buffer. The result is always null terminated.
Signature Details	cin >> buf;

Examples of Incorrect Code	`[...]`
	`char user_name[10];`
	`cin >> user_name;`
	`[...]`

Examples of Corrected Code	`[...]`
	`char user_name[10];`
	`cin.width(10);`
	`cin >> user_name;`
	`[...]`

Source Reference	• http://www.cplusplus.com/ref/iostream/
Recommended Resource	Howard, Michael & LeBlanc, David C. *Writing Secure Code, 2nd ed*. Redmond, WA: Microsoft Press, 2002, Appendix A, ISBN: 0735617228.
Discriminant Set	**Operating System**: • Windows **Language**: • C++

1. mailto:copyright@cigital.com